

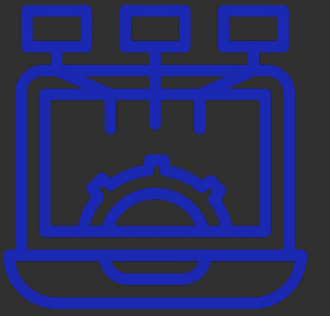
Cool Cyber Games

Interactive Web Application for
Cybersecurity Learning



Team: Matthew Goembel, Anthony Clayton, Ludendorff Brice, Ben Allerton

Milestone 1 Overview



Select Collaboration Tools



Create Design Doc



Select Technical
Tools



Create Test Plan Doc



Resolve Technical
Challenges*



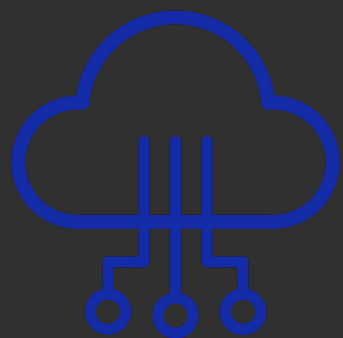
Create "hello world"
demos



Create Requirements
Doc



Milestone 2



01

Compare and Select Tools

Tools



■ Collaborative:

- Github
- Jira
- Discord
- Drive
- Email/Message

■ Technical:

- Front-end: HTML, CSS, JavaScript
- Backend: Node.js, Render, and Express.js
- Database: MongoDB (separate databases for User DB and Game DB)
- Game Engine: GDScript(Godot)
- Authentication: OAuth 2.0 (Google integration)

Integration: Google APIs

02

Resolving Technical Challenges

Continued Challenges



- **Bandwidth Concerns** - Handle site traffic efficiently and cost-effectively
- **Implementing API Connections** - Design an accurate system to track user stats
from games
- **Security** - Maintaining site and data security
- **Effective Learning Content*** - Continuing to evolve content to maximize learning

03

Requirements

Requirements Focus



■ Purpose:

- A web application to teach cybersecurity concepts interactively.
- Hands-on learning through gamified cybersecurity challenges.

■ Scope:

- Platform and Learning functionality
- Gamified challenges & tutorials
- Progress tracking & certifications
- Cross-platform accessibility

Functional



- **User Authentication** – Google OAuth integration
- **Interactive Tutorials & Quizzes** – Multiple formats (MCQs, drag-and-drop, scenarios)
- **Gamified Learning** – Real-world cybersecurity threats, levels, hints
- **Progress Tracking** – Leaderboards, certificates
- **User Dashboard** – Personalized stats & progress

Non-Functional / Interface



- **Performance:** Supports 100+ users, <2s page load time
- **Security:** HTTPS, OWASP best practices
- **Usability:** ADA-compliant, intuitive UI
- **Scalability:** Expandable features, database growth

04

Design

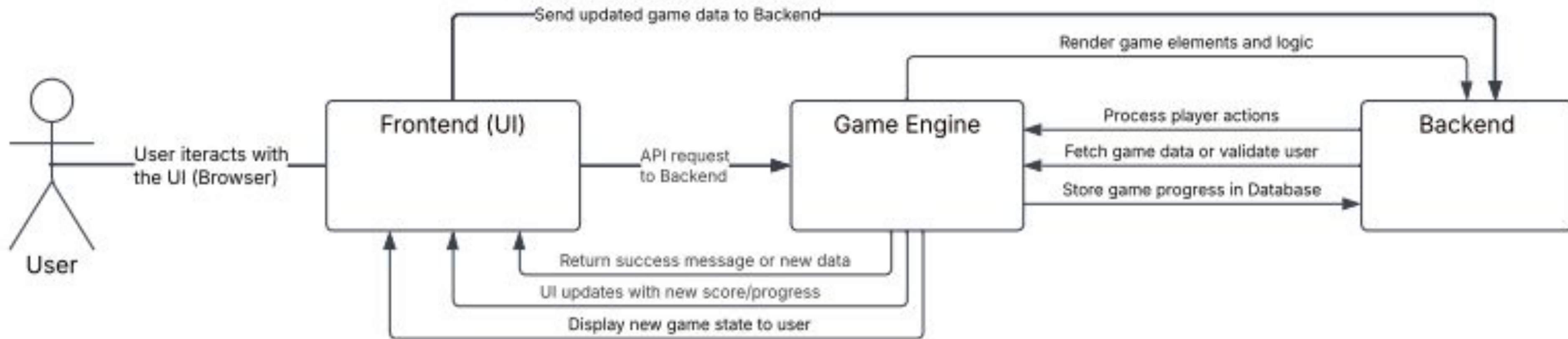
Overview



■ The system follows a modular, service-oriented architecture consisting of the following components:

- Frontend Module
- Backend Module
- Database Module
- Game Engine
- Authentication Service
- User Data Tracking Service

System Architecture



Current UI



Cool Cyber Games

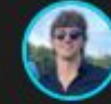
[Home](#)

[Games](#)

[Dashboard](#)

[About](#)

[Contact](#)



Matthew Goembel

Welcome to Cool Cyber Games

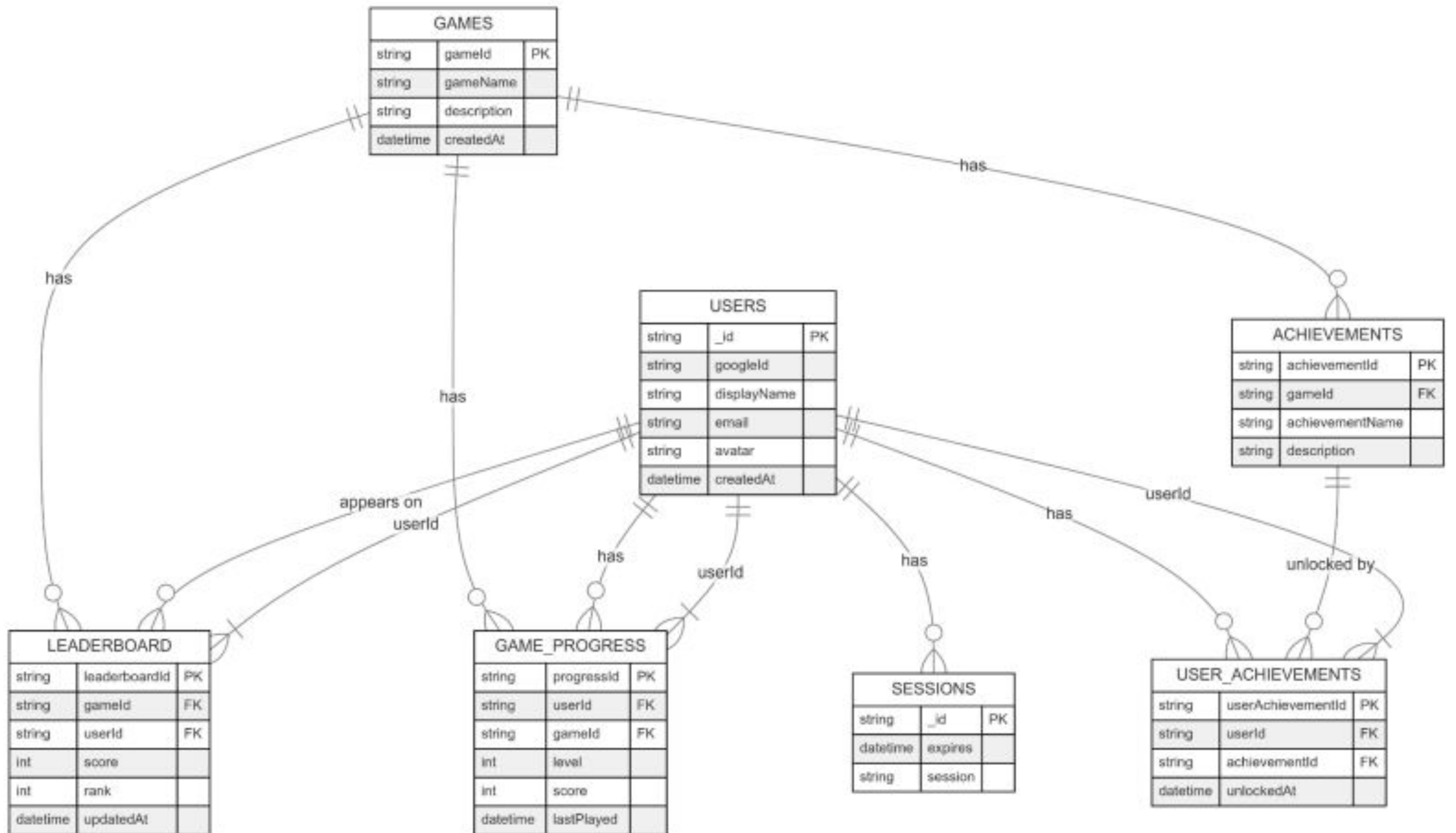
Embark on an epic journey through our cybersecurity challenges. Learn, compete, and become a master of digital defense through interactive gameplay and real-world scenarios.

[Start Your Journey](#)

Featured Games



Malware Maze



05

Test Plan

Overview / Scope



■ Key areas:

- Functional Testing (core features, tutorials, challenges)
- Performance Testing (load, response times)
- Security Testing (OAuth 2.0, SQL Injection, XSS protection)
- Usability and Compatibility Testing (cross-device, browsers)

Key Test Areas



■ User Authentication

- Google OAuth 2.0 Login & Logout
- Token Expiry and Refresh

■ Interactivity & Progress Tracking

- Access and complete interactive tutorials
- Quiz results, points, and progress tracking

■ Cybersecurity Challenges

- Phishing Email Identification
- Malware file removal
- Social Engineering
- etc.

Server & Database Testing



- **Database:** MongoDB (User & Game Data)

- **Test Cases:** Connection, Performance, Backup, and Recovery

- **Server:** Hosted on Render

- **Test Cases:** Uptime, API Endpoints, Scalability, Security

06

"hello world" demo

→ <https://coolcybergames.com>

07

Milestone 2

Next Tasks



■ 1. Core Features Implementation:

- Develop APIs for user authentication, progress tracking, and points.
- More games, Quiz scoring, Progress UI, Global score tracking

■ 2. Gamification and Authentication Enhancements:

- Add interactive elements, quiz scoring logic, and Google login.
- Develop progress-tracking/dashboard UI.
- Focus on testing for performance and security.

Next Tasks



- **3. Continued Frontend & Game Development:**
 - Add and update web pages progressively.
 - Implement and improve game functionality on the site.
 - Enhance website functionality for seamless user experience



Thank you!

Any Questions?

